

An asymptotically optimal Bernoulli factory for certain functions that can be expressed as power series

Luis Mendo

*E.T.S. Ingenieros de Telecomunicación, Technical University of Madrid,
28040 Madrid, Spain. e-mail: luis.mendo@upm.es*

Abstract: Given a sequence of independent Bernoulli variables with unknown parameter p , and a function f that can be expressed as a power series with non-negative coefficients, an algorithm is presented that produces a Bernoulli random variable with parameter $f(p)$. In particular, the algorithm can simulate $f(p) = p^a$ for $a \in (0, 1)$. For the subclass of functions f that are asymptotically proportional to p^a as $p \rightarrow 0$, the algorithm requires an average number of inputs that is asymptotically optimal in a precisely defined sense. A non-randomized version of the algorithm is also given. The distribution of the number of inputs required by any of these algorithms has an exponentially decaying tail. Some extensions of the algorithms are discussed.

MSC 2010 subject classifications: 65C50.

Keywords and phrases: Bernoulli factory, Simulation, Power series.

1. Introduction

Let $X = (X_i)_{i \in \mathbb{N}}$ denote a sequence of independent, identically distributed (i.i.d.) Bernoulli random variables X_i with unknown parameter p . A *non-randomized stopping rule* on X is a sequence of *stopping functions* $\tau_1(x_1)$, $\tau_2(x_1, x_2)$, \dots , $\tau_i(x_1, \dots, x_i)$, \dots that take values in $\{0, 1\}$, such that for almost all realizations x_1, x_2, \dots of X at least one of the functions τ_i takes the value 1. The *stopping time* N is the smallest integer i for which this occurs, or infinity if it does not occur.

A *randomized stopping rule* uses, in addition to X , a *randomizing sequence* $U = (U_i)_{i \in \mathbb{N}}$ of independent random variables with uniform distribution on $(0, 1)$. The stopping function τ_i depends on $x_1, u_1; x_2, u_2; \dots; x_i, u_i$, where each u_j represents the value taken by the random variable U_j . The stopping time N is defined as before, and is also assumed to be finite almost surely. In addition, it is assumed that the functions τ_i are measurable with respect to the product measure obtained from using counting measure for the x_j variables and Lebesgue measure for the u_j variables.

Another possible definition of a randomizing stopping rule would be to specify that at each i , given $X_1 = x_1, \dots, X_i = x_i$, there is a certain probability of stopping that depends on x_1, \dots, x_i . This corresponds to the above definition if the output of τ_i is obtained from comparing U_i with a threshold that depends on x_1, \dots, x_i . Thus the definition based on the auxiliary sequence U captures all the randomness that can be effected by a randomized stopping rule (and is more convenient for the purposes of this paper).

Let $f : S \rightarrow (0, 1)$ denote a function defined on a set $S \subseteq (0, 1)$. A *non-randomized Bernoulli factory* or *simulation* of f based on X is an algorithm that, using values from the sequence X as inputs, generates a Bernoulli variable Y with parameter

$f(p)$. Specifically, the number N of required inputs is a stopping time on X dictated by a non-randomized stopping rule; and the output value Y depends on X_1, \dots, X_N .

A *randomized Bernoulli factory* uses, in addition to the sequence X , an auxiliary sequence U of independent random variables uniformly distributed on $(0, 1)$. Specifically, N is given by a randomized stopping rule on X with U as randomizing sequence. The output Y is also possibly randomized, that is, it depends on X_1, \dots, X_N and U_1, \dots, U_N . Specifically, there exists a sequence of functions $\gamma_1, \gamma_2, \dots$ such that for $N = n$ the output is given as $\gamma_n(x_1, u_1; \dots; x_n, u_n)$. These functions are assumed to be measurable with respect to the product measure referred to above.

One of the earliest references of a Bernoulli factory, for the specific case $f(p) = 1/2$, appears in a work by [von Neumann \(1951\)](#). For general f , [Keane & O'Brien \(1994\)](#) give a necessary and sufficient condition for a simulation of f to be possible, namely that the function either is constant, or is continuous and satisfies a certain polynomial bound. [Nacu & Peres \(2005\)](#) define a non-randomized simulation to be *fast* if the distribution of N has an exponential tail, that is, for any $p \in S$ there exist values $A > 0$, $\beta < 1$ (which may depend on p) such that

$$\Pr[N > n] \leq A\beta^n. \quad (1.1)$$

The authors prove that a fast simulation exists for any f real analytic on any closed interval $I \subset (0, 1)$.

Several works on Bernoulli factories present simulation algorithms for specific functions. Considerable attention has been given to $f(p) = \min\{2p, 1 - 2\varepsilon\}$, $\varepsilon > 0$, which is an important building block for simulating other functions ([Łatuszyński et al., 2011](#); [Nacu & Peres, 2005](#)); and to linear functions $f(p) = cp$, $c > 1$ defined on a suitable set $S \subset (0, 1/c)$; see [Huber \(2016\)](#) and references therein.

A crucial parameter of a Bernoulli factory is $E[N]$, that is, how many inputs X_i are required on average to generate a sample of Y . In applications, observing the variables X_i is usually costly, and thus $E[N]$ should be as small as possible. For randomized Bernoulli factories, the auxiliary random variables U_i are assumed to be cost-free.

This paper deals mainly with functions of the form $f(p) = 1 - \sum_{k=1}^{\infty} c_k(1-p)^k$ where the coefficients c_k are non-negative and sum to 1. A randomized algorithm to simulate any such function is presented in Section 2, and its average number of inputs, $E[N]$, is computed. The algorithm can be particularized to functions $f(p) = p^a$, $a \in (0, 1)$. For $a = 1/2$ the algorithm is similar to that given by [Wästlund \(1999\)](#); and the presented results affirmatively answer question 1 by [Nacu & Peres \(2005\)](#), i.e. establish that \sqrt{p} can be simulated with finite $E[N]$.

An interesting subclass is that of functions that asymptotically behave like bp^a , $a \in (0, 1)$, $b \in (0, \infty)$ as $p \rightarrow 0$. For these functions, it will be seen that $E[N]$ grows without bound as $p \rightarrow 0$, no matter what simulation algorithm be used. Therefore it is important to analyse the asymptotic rate of growth of $E[N]$. Furthermore, the case of vanishingly small p is of practical interest, because in many applications the probability values are indeed small. This analysis is presented in Section 3. The results show that the proposed algorithm is asymptotically optimal for the mentioned subclass of functions, in the sense that for any other algorithm $E[N]$ grows at least as fast.

A non-randomized version of the proposed algorithm is given in Section 4, and is shown to be fast in the sense of [Nacu & Peres \(2005\)](#). Section 5 discusses some extensions of the algorithms to cover a broader range of functions. Section 6

presents conclusions and open problems. Section 7 contains proofs to all results.

The following notation is used throughout the paper. $x^{(m)}$ represents the rising factorial $x(x+1)\cdots(x+m-1)$ for $m \in \mathbb{N}$, and $x^{(0)} = 0$. $(x)_m$ is the falling factorial $x(x-1)\cdots(x-m+1)$, $(x)_0 = 0$. Given two positive functions f_1 and f_2 , $f_1(x)$ is said to be $\Omega(f_2(x))$ for $x \rightarrow x_0$ if there is a constant $C > 0$ such that $f_1(x) \geq Cf_2(x)$ for all x sufficiently close to x_0 . Similarly, $f_1(x)$ is $\Theta(f_2(x))$ for $x \rightarrow x_0$ if there exist constants $C, C' > 0$ such that $C'f_2(x_0) \geq f_1(x_0) \geq Cf_2(x_0)$ for x all sufficiently close to x_0 . $f_1(x)$ is $\omega(f_2(x))$ for $x \rightarrow x_0$ if $f_1(x)/f_2(x) \rightarrow \infty$ as $x \rightarrow x_0$.

2. Simulation algorithm. Average number of inputs

Consider an i.i.d. sequence X of random variables X_i that take the value 1 with probability p and 0 with probability $1-p$. Let $f : (0, 1) \rightarrow (0, 1)$ be a function that can be expressed as a power series

$$f(p) = 1 - \sum_{k=1}^{\infty} c_k (1-p)^k \quad (2.1)$$

with

$$c_k \geq 0, \quad (2.2)$$

$$\sum_{k=1}^{\infty} c_k = 1. \quad (2.3)$$

Note that this implies that $\lim_{p \rightarrow 0} f(p) = 0$ and $\lim_{p \rightarrow 1} f(p) = 1$.

The randomized algorithm to be presented yields a Bernoulli random variable Y with parameter $f(p)$. It takes as inputs a number of random variables from the sequence X , as well as from an auxiliary sequence U of i.i.d. random variables U_i uniformly distributed on $(0, 1)$ and independent from the X_i variables. The algorithm makes use of coefficients d_k computed from c_k as follows:

$$d_k = \frac{c_k}{1 - \sum_{j=1}^{k-1} c_j}. \quad (2.4)$$

From (2.2) and (2.3) it stems that $0 \leq d_k \leq 1$. If the number of non-zero coefficients c_k is finite, i.e. there exists K such that $c_K > 0$ and $c_k = 0$ for $k > K$, (2.4) gives $d_K = 1$; and for $k > K$ the coefficient d_k is not defined (and is not necessary, as will be seen).

Algorithm 2.1. Let f be given by (2.1)–(2.3), and let d_k be defined by (2.4). The input of the algorithm is a sequence X of i.i.d. Bernoulli random variables. The output is a Bernoulli random variable Y .

1. Set $i = 1$.
2. Take one input X_i .
3. Produce U_i uniform on $(0, 1)$. Let $V_i = 1$ if $U_i < d_i$ or $V_i = 0$ otherwise.
4. If V_i or X_i are 1, output $Y = X_i$ and finish. Else increase i and go back to step 2.

The idea of this algorithm is roughly similar to that presented by Wästlund (1999) to simulate $f(p) = \sqrt{p}$, namely, decompose the event $Y = 0$ as an infinite

sum of mutually exclusive events, each with probability $c_k(1-p)^k$. However, there are two differences here. First, the referenced paper treats the c_k and $(1-p)^k$ parts separately. Namely, an auxiliary random variable L is first generated with $\Pr[L = k] = c_k$, $k \in \mathbb{N}$. This variable represents the amount of inputs X_i that need to be taken. Then, $(1-p)^L$ is simulated using the product $\prod_{i=1}^L (1-X_i)$. On the other hand, Algorithm 2.1 reduces the number of required inputs X_i by stopping as soon as one of the X_i variables is 1. This can be done because in that case the above product is 0 irrespective of the values of the remaining X_i variables.

The second difference is that Algorithm 2.1 uses auxiliary Bernoulli variables V_i with respective parameters d_i , instead of a random variable L with the distribution given by coefficients c_k . Using the Bernoulli variables is simpler and more general. The algorithm given by Wästlund (1999) can generate L directly because, for the specific case $f(p) = \sqrt{p}$, the coefficients c_k in (2.1) are

$$c_k = \frac{1}{2^{2k-1}k} \binom{2k-2}{k-1}. \quad (2.5)$$

As it turns out, if a fair coin is flipped until the total number of tails exceeds the total number of heads, the probability that this happens after $2k-1$ steps is precisely (2.5). It is this fact that allows generating L in a simple manner for $f(p) = \sqrt{p}$. But for other functions there may not be an analogous procedure. The Bernoulli random variables V_i provide a good alternative, which can be used for arbitrary coefficients c_k . Effectively, each d_k represents the conditional probability that $L = k$ given that $L \geq k$. The following proposition makes this clear.

Proposition 2.1. *The coefficients d_k defined by (2.4) satisfy*

$$c_k = d_k \prod_{j=1}^{k-1} (1-d_j). \quad (2.6)$$

Additionally, from this interpretation it stems that if the series in (2.1) is finite, with $c_K > 0$, $c_k = 0$ for k exceeding K , it is unnecessary to define coefficients d_k for $k > K$, because $d_K = 1$.

Theorem 2.1. *Given a sequence X of i.i.d. Bernoulli random variables with parameter $p \in (0, 1)$, a function f of the form (2.1)–(2.3), and coefficients d_k computed from (2.4), the Bernoulli random variable Y produced by Algorithm 2.1 has $\Pr[Y = 1] = f(p)$.*

Algorithm 2.1 takes a new input X_i at each step i . Thus the number of used inputs, N , coincides with the number of steps required by the algorithm.

Theorem 2.2. *For f given by (2.1)–(2.3) and $p \in (0, 1)$, the average number of inputs required by Algorithm 2.1 is*

$$\mathbb{E}[N] = \frac{f(p)}{p}. \quad (2.7)$$

In addition, the distribution of N has an exponential tail.

It is interesting to consider the case $f(p) = p^a$, $a \in (0, 1)$. These functions can be expressed in the form (2.1) with

$$c_k = \frac{(1-a)^{(k-1)}a}{k!}, \quad (2.8)$$

from which $d_k = a/k$. Algorithm 2.1 can be applied, and $E[N] = p^{a-1}$. In particular, $f(p) = \sqrt{p}$ can be simulated with $E[N] = 1/\sqrt{p}$. This solves Question 1 of Nacu & Peres (2005), which asks if there is an algorithm that simulates \sqrt{p} on $(0, 1)$ for which the number of required inputs has finite expectation for all p .

3. Asymptotic optimality

A natural question is whether the average number of inputs required by Algorithm 2.1 can be improved by some other algorithm. The most interesting case is that of functions f for which Algorithm 2.1 gives $E[N] \rightarrow \infty$ as $p \rightarrow 0$; that is, functions that are $\omega(p)$ at the origin. In this case, since the average number of inputs used by the algorithm grows without bound, it is important to know if the growth rate could be reduced by using some other algorithm. As will be seen in Theorem 3.1, for a certain subclass of these functions, the average number of inputs required by any algorithm increases for $p \rightarrow 0$ at least as fast as it does with Algorithm 2.1, which is thus asymptotically optimal.

The proof of this result is based on the following idea. Consider an arbitrary, possibly randomized algorithm for generating a Bernoulli variable with parameter $s = f(p)$ from the sequence X . Let N denote the random number of inputs required by this algorithm. Given $t \in \mathbb{N}$, apply the algorithm repeatedly to yield t i.i.d. values Y_j , $j = 1, \dots, t$. From these estimate the parameter s using the standard, fixed-size estimator $\hat{s} = \sum_{j=1}^t Y_j / t$.

The above process can be seen as a randomized sequential estimation of $f(p)$ based on X . The number M of used inputs is a randomized stopping time on the sequence X , with $E[M] = t E[N]$. Wolfowitz's extension of the Cramér-Rao bound to sequential estimators (Wolfowitz, 1947) can be applied to \hat{s} , which sets a lower bound on $\text{Var}[\hat{s}]$. Computing the actual variance and combining with this gives a lower bound on $E[N]$, from which the claimed asymptotic result is obtained.

Consider the class of functions $f : (0, 1) \rightarrow (0, 1)$ for which

$$\lim_{p \rightarrow 0} \frac{f(p)}{p^a} = b \quad \text{for some } a \in (0, 1), b \in (0, \infty). \quad (3.1)$$

This condition implies that $\lim_{p \rightarrow 0} f(p) = 0$. If f is differentiable, applying L'Hôpital's rule it stems that (3.1) is equivalent to the following condition on the derivative f' :

$$\lim_{p \rightarrow 0} f'(p) p^{1-a} = ab \quad (3.2)$$

where a and b are the same values as in (3.1).

The functions that fulfill (3.1) in addition to (2.1)–(2.3) form a strict subset of the class of functions defined by (2.1)–(2.3). Consider for example $f : (0, 1) \rightarrow (0, 1)$ given as

$$f(p) = p(1 - \log p) = 1 - \sum_{k=2}^{\infty} \frac{1}{k(k-1)} (1-p)^k. \quad (3.3)$$

Identifying terms in (2.1) and (3.3), it is clear that (2.2) and (2.3) are satisfied. However, this function has $f'(p) = -\log p$ and does not satisfy (3.2), because $\lim_{p \rightarrow 0} p^{1-a} \log p = 0$ for any $a \in (0, 1)$. On the other hand, for $a \in (0, 1)$ the function $f(p) = p^a$ clearly satisfies condition (3.1) with $b = 1$.

Theorem 3.1. *If a function $f : (0, 1) \rightarrow (0, 1)$ given by (2.1)–(2.3) satisfies condition (3.1), any (possibly randomized) Bernoulli factory for this function uses an average number of inputs $E[N] = \Omega(f(p)/p)$ for $p \rightarrow 0$.*

Corollary 3.1. *For any function f given by (2.1)–(2.3) that satisfies condition (3.1), or equivalently (3.2), Algorithm 2.1 is asymptotically optimal as $p \rightarrow 0$; that is, for any algorithm, with average number of inputs $E[N]$, there exist $C, \varepsilon > 0$ such that $E[N] \geq Cf(p)/p$ for all $p < \varepsilon$.*

4. Non-randomized algorithm

A non-randomized version of Algorithm 2.1 is given below. Instead of using an auxiliary variable U_i to produce a Bernoulli variable V_i with parameter d_i in step 3, the required V_i is obtained from additional input samples using the following well known procedure (see for example proposition 13 of [Nacu & Peres \(2005\)](#)).

Algorithm 4.1. *The algorithm uses the same steps 1–4 from Algorithm 2.1 except that step 3 is replaced by the following:*

- 3.1. Set $j = 1$.
- 3.2. Keep taking pairs of values from the sequence X until the two values of a pair are different. Let T be the first value of that pair.
- 3.3. If $T = 0$ increase j and go back to step 3.2. Else set V_i equal to the j -th digit in the fractional part of the binary expansion of d_i .

The total number of inputs taken from X is obviously greater than with Algorithm 2.1. However, the final value of i in Algorithm 2.1 has an exponential tail, which can be used for establishing that Algorithm 4.1 is fast in the sense of [Nacu & Peres \(2005\)](#).

Theorem 4.1. *Algorithm 4.1 requires an average total number of inputs*

$$E[N] = \frac{f(p)}{p} \left(1 + \frac{2}{p(1-p)} \right). \quad (4.1)$$

In addition, the algorithm is fast; that is, the distribution of N has an exponential tail.

5. Extensions of the algorithms

The presented algorithms can be modified in several ways to extend the range of functions that can be simulated. Algorithm 2.1 will be considered in the following, but the discussion also applies to its non-randomized version given by Algorithm 4.1.

An obvious modification is to change step 4 of the algorithm so that instead of $Y = X_i$ it outputs the complementary variable $Y = 1 - X_i$. This simulates the function $g(p) = 1 - f(p)$. The average number of inputs and asymptotic optimality of the modified algorithm are unaffected. Equation (2.1) and condition (3.1) are respectively replaced by $g(p) = \sum_{k=1}^{\infty} c_k(1-p)^k$ and $\lim_{p \rightarrow 0} (1 - g(p))/p^a = b$ for some $a \in (0, 1)$, $b \in (0, \infty)$; whereas (2.2) and (2.3) are maintained.

The same operation can be applied to the input variables in step 2. This allows simulation of functions $g(p) = f(1 - p)$, where f satisfies the conditions of the original algorithm; and the simulation is asymptotically optimal for $p \rightarrow 1$.

It is also possible to simulate a function obtained from applying certain operations to two constituent functions. Consider $f_1(p) = 1 - \sum_{k=1}^{\infty} c_{1,k}(1-p)^k$ and $f_2(p) = 1 - \sum_{k=1}^{\infty} c_{2,k}(1-p)^k$. Define the functions f (composition), g (product with complement) and h (convex combination) as follows:

$$f(p) = f_2(f_1(p)) \quad (5.1)$$

$$g(p) = 1 - (1 - f_1(p))(1 - f_2(p)) \quad (5.2)$$

$$h(p) = \alpha f_1(p) + (1 - \alpha)f_2(p) \quad \text{for } \alpha \in (0, 1). \quad (5.3)$$

Proposition 5.1. *Functions f , g and h satisfy (2.1)–(2.3) and (3.1) provided that f_1 and f_2 do.*

By Proposition 5.1, Algorithm 2.1 can be used to simulate f , g or h . Alternatively, it is possible to simulate f_1 and f_2 separately and then combine the results to obtain the desired function. In the three cases this alternative approach is easily seen to require the same average number of inputs as applying Algorithm 2.1 to f , g or h . Consider for example the case of function f . In the alternative approach, Algorithm 2.1 is first applied to simulate f_1 with input sequence X . This produces a sequence of Bernoulli variables with parameter $f_1(p)$. Then the algorithm is applied again to simulate f_2 on this sequence. The first stage requires $f_1(p)/p$ inputs on average. The second uses on average $f_2(f_1(p))/f_1(p)$ outputs of the first stage as inputs. The average number of original inputs is the product of those two numbers, which equals $f(p)/p$.

Of course, other combinations of functions may be realizable, even if the resulting function cannot be simulated directly by a single application of the algorithm. For example, if f_1, f_2 satisfy the conditions for Algorithm 2.1, it is immediate to simulate $f(p) = f_1(p)f_2(p)$ by multiplying the outputs for f_1 and f_2 (with “short-circuiting” to reduce the average number of required inputs). However, it may not be possible to simulate f directly because its coefficients c_k are not necessarily non-negative.

6. Conclusions and future work

An algorithm has been presented that can simulate certain functions f using an average number of inputs that is asymptotically optimal for p vanishingly small. This algorithm generalizes that given by Wästlund (1999) for $f(p) = \sqrt{p}$, uses fewer inputs, and admits a non-randomized version that is fast in the sense of Nacu & Peres (2005).

In future research, it would be interesting to relax the sufficient conditions for asymptotic optimality (Theorem 3.1), and to extend the algorithm to a more general class of functions, especially regarding condition (2.2).

7. Proofs

7.1. Proof of Proposition 2.1

From (2.3) and (2.4) it stems that

$$d_k = \frac{c_k}{\sum_{j=k}^{\infty} c_j}, \quad (7.1)$$

$$1 - d_k = \frac{\sum_{j=k+1}^{\infty} c_j}{\sum_{j=k}^{\infty} c_j}. \quad (7.2)$$

Combining (7.1) and (7.2),

$$d_k \prod_{j=1}^{k-1} (1 - d_j) = \frac{c_k}{\sum_{j=k}^{\infty} c_j} \frac{\sum_{j=2}^{\infty} c_j}{\sum_{j=1}^{\infty} c_j} \frac{\sum_{j=3}^{\infty} c_j}{\sum_{j=2}^{\infty} c_j} \cdots \frac{\sum_{j=k}^{\infty} c_j}{\sum_{j=k-1}^{\infty} c_j} = \frac{c_k}{\sum_{j=1}^{\infty} c_j} = c_k. \quad (7.3)$$

which completes the proof. \square

7.2. Proof of Theorem 2.1

The algorithm ends at step n producing output $Y = 0$ if and only if $X_i = 0$, $V_i = 0$ for $i \geq n-1$; $V_n = 1$; and $X_n = 0$. Since all these variables are independent,

$$\Pr[N = n, Y = 0] = (1 - d_1)(1 - d_2) \cdots (1 - d_{n-1})d_n(1 - p)^n, \quad (7.4)$$

which according to (2.6) equals $c_n(1 - p)^n$. Therefore

$$\Pr[Y = 1] = 1 - \sum_{n=1}^{\infty} \Pr[N = n, Y = 0] = 1 - \sum_{n=1}^{\infty} c_n(1 - p)^n = f(p). \quad (7.5)$$

as desired. \square

7.3. Proof of Theorem 2.2

The algorithm requires at least n steps if and only if $X_i = 0$, $V_i = 0$ for $i \leq n-1$; that is,

$$\Pr[N \geq n] = (1 - d_1)(1 - d_2) \cdots (1 - d_{n-1})(1 - p)^{n-1}. \quad (7.6)$$

Thus

$$\mathbb{E}[N] = \sum_{n=1}^{\infty} \Pr[N \geq n] = \sum_{n=1}^{\infty} (1 - d_1)(1 - d_2) \cdots (1 - d_{n-1})(1 - p)^{n-1}. \quad (7.7)$$

Making use of (2.4) and (2.6),

$$\mathbb{E}[N] = \sum_{n=1}^{\infty} \frac{c_n}{d_n} (1 - p)^{n-1} = \sum_{n=1}^{\infty} (1 - p)^{n-1} \left(1 - \sum_{j=1}^{n-1} c_j \right) = \frac{1}{p} - \sum_{n=1}^{\infty} \sum_{j=1}^{n-1} c_j (1 - p)^{n-1}. \quad (7.8)$$

Since all the terms in the double series are non-negative, the order of summation can be changed. This gives, taking into account (2.1),

$$\mathbb{E}[N] = \frac{1}{p} - \sum_{j=1}^{\infty} \sum_{n=j+1}^{\infty} c_j (1 - p)^{n-1} = \frac{1}{p} - \sum_{j=1}^{\infty} \frac{c_j (1 - p)^j}{p} = \frac{f(p)}{p}. \quad (7.9)$$

Since all coefficients d_i are upper-bounded by 1, from (7.6) it stems that

$$\Pr[N > n] = (1 - d_1)(1 - d_2) \cdots (1 - d_n)(1 - p)^n \leq (1 - p)^n, \quad (7.10)$$

and thus (1.1) holds with $A = 1$, $\beta = 1 - p < 1$. \square

7.4. Proof of Theorem 3.1

Let $f : (0, 1) \rightarrow (0, 1)$ be a function defined by (2.1)–(2.3) that satisfies condition (3.1). Consider an arbitrary, randomized Bernoulli factory \mathcal{B} for f based on the sequence X with parameter p , randomizing sequence U , and stopping functions $\tau_i(x_1, u_1; \dots, x_i, u_i)$, $i \in \mathbb{N}$, assumed to be measurable.

The randomized Bernoulli factory \mathcal{B} can be replaced by an equivalent, non-randomized sequential procedure \mathcal{B}' that produces the same output using an input sequence Z defined by $Z_i = X_i + U_i$. The equivalence is clear from the fact that X_i and U_i can be retrieved from Z_i as $X_i = \lfloor Z_i \rfloor$, $U_i = Z_i - \lfloor Z_i \rfloor$. The stopping functions of \mathcal{B}' , denoted by τ'_i , are related to those of \mathcal{B} as $\tau'_i(z_1, \dots, z_i) = \tau_i(\lfloor z_1 \rfloor, z_1 - \lfloor z_1 \rfloor; \dots; \lfloor z_i \rfloor, z_i - \lfloor z_i \rfloor)$. Each τ'_i is measurable because τ_i is. Let the random variable N represent the number of Z_i inputs used by \mathcal{B}' (or of X_i inputs used by \mathcal{B}). The sequence Z will be said to have parameter p if the underlying X sequence has parameter p .

For $p \in (0, 1)$, let $s = f(p)$, $t = \lfloor 1/s \rfloor$. Consider a sequential procedure \mathcal{P} that consists in applying t times the equivalent Bernoulli factory \mathcal{B}' to inputs taken from Z with parameter p , which produces t i.i.d. Bernoulli variables Y_1, \dots, Y_t with parameter $s = f(p)$. The number of X_i inputs used by \mathcal{P} will be denoted as M , and is given by $N_1 + \dots + N_t$, where N_j is the number of inputs used by the j -th application of \mathcal{B}' . The stopping time M is randomized from the point of view of X , but is non-randomized with respect to Z . Let \hat{s} denote the standard estimator of s from Y_1, \dots, Y_t ,

$$\hat{s} = \frac{\sum_{j=1}^t Y_j}{t}. \quad (7.11)$$

For $M = m$, the random variables Z_1, \dots, Z_m are independent; and their joint probability density, with respect to Lebesgue measure, is $\lambda_m(z_1, \dots, z_m) = \prod_{i=1}^m \lambda(z_i)$ with

$$\lambda(z) = \begin{cases} 1 - p & \text{if } z \in (0, 1) \\ p & \text{if } z \in (1, 2). \end{cases} \quad (7.12)$$

Let $R_m \subseteq (0, 2)^m$ be the set of m -tuples (Z_1, \dots, Z_m) that give $M = m$ (that is, such that \mathcal{P} stops at the m -th step). Since the stopping functions τ'_i are measurable, the region R_m is measurable too; and

$$\Pr[M = m] = \int_{R_m} \lambda_m(z_1, \dots, z_m) dz_1 \cdots dz_m. \quad (7.13)$$

For $M = m$ the estimator \hat{s} defined by (7.11) can be expressed as a function ζ_m that depends only on Z_1, \dots, Z_m (the random variables Z_1, \dots, Z_m determine the values Y_j in the numerator of (7.11) through the functions γ_i). In addition, let

$$\varphi_m(p) = \mathbb{E}[\hat{s} \mid M = m] \Pr[M = m] = \int_{R_m} \zeta_m(z_1, \dots, z_m) \lambda_m(z_1, \dots, z_m) dz_1 \cdots dz_m. \quad (7.14)$$

The following two lemmas will be used in the proof of the theorem.

Lemma 7.1. *If $E[M]$ is finite, the series $\sum_{m=1}^{\infty} \partial \phi_m(p) / \partial p$ converges uniformly on any interval $p \in (\varepsilon, 1 - \varepsilon)$, $\varepsilon > 0$.*

Proof. It will be proved first that differentiation under the integral sign is possible in (7.14). This requires checking certain regularity conditions, so that Leibniz's rule can be applied. Specifically (Fleming, 1977, page 237), it suffices to show that the subintegral function and its derivative are continuous and bounded.

Let R_m be divided into 2^t disjoint sub-regions, each associated with a tuple (y_1, \dots, y_t) of possible values for the variables Y_1, \dots, Y_t . Namely, $(z_1, \dots, z_m) \in R_m$ if and only for $Z_1 = z_1, \dots, Z_m = z_m$ the sequential procedure \mathcal{P} stops at $M = m$ producing outputs $Y_1 = y_1, \dots, Y_t = y_t$. Each of these subregions will be denoted as $R_{m, (y_1, \dots, y_t)}$. Similarly, $R_{n, y}$ for $y \in \{0, 1\}$ will denote the region of values of z_1, \dots, z_n such that a single run of \mathcal{B}' with $Z_1 = z_1, \dots, Z_n = z_n$ stops at the n -th step producing the output $Y = y$. A given m can correspond to different values n_1, \dots, n_t for the numbers of inputs taken by the t runs of \mathcal{B}' , as long as $n_1 + \dots + n_t = m$. This means that the region $R_{m, (y_1, \dots, y_m)}$ is obtained as a union of disjoint sets,

$$R_{m, (y_1, \dots, y_t)} = \bigcup_{n_1, \dots, n_t} R_{n_1, y_1} \times \dots \times R_{n_t, y_t} \quad (7.15)$$

where the union is extended to all $n_1, \dots, n_t \in \{1, \dots, m - t + 1\}$ such that $n_1 + \dots + n_t = m$.

The set of tuples (z_1, \dots, z_n) such that \mathcal{B}' stops at step n is defined by the conditions $\tau'_1(z_1) = 0$, $\tau'_2(z_1, z_2) = 0, \dots, \tau'_n(z_1, \dots, z_n) = 1$. Since the stopping functions τ'_i are measurable, the referred set is measurable. Similarly, the function ζ_n is also measurable because the functions γ_i are; and thus $\zeta_n^{-1}(\{y\})$ is a measurable set for $y \in \{0, 1\}$. The intersection of these two sets gives the region $R_{n, y}$, which is thus measurable; and in view of (7.15) so is $R_{m, (y_1, \dots, y_t)}$.

Defining $r = \lfloor z_1 \rfloor + \dots + \lfloor z_m \rfloor$ allows writing $\lambda_m(z_1, \dots, z_m)$ as $p^r (1 - p)^{m-r}$; and

$$\frac{\partial \lambda_m(z_1, \dots, z_m)}{\partial p} = (r - mp) p^r (1 - p)^{m-r} = \frac{r - mp}{p(1 - p)} \lambda_m(z_1, \dots, z_m) \quad (7.16)$$

This function has a discontinuity when any z_i approaches 1 (which causes a jump in r), but is continuous within each of the 2^m hypercubes $(x_1, x_1 + 1) \times \dots \times (x_m, x_m + 1)$, $x_i \in \{0, 1\}$, $i = 1, \dots, m$. These hypercubes will be denoted as H_{x_1, \dots, x_m} . In order to differentiate under the integral sign in (7.14) each of the sub-regions $R_{m, (y_1, \dots, y_t)}$ needs to be further divided into 2^m sets resulting from its intersection with one of the hypercubes H_{x_1, \dots, x_m} . The 2^{t+m} resulting sets are disjoint and measurable with respect to Lebesgue measure. The integral in (7.14) can thus be expressed as

$$\phi_m(p) = \sum_{x_1, \dots, x_m, y_1, \dots, y_t} \int_{H_{x_1, \dots, x_m} \cap R_{m, (y_1, \dots, y_t)}} \zeta_m(z_1, \dots, z_m) \lambda_m(z_1, \dots, z_m) dz_1 \dots dz_m \quad (7.17)$$

where the sum is has 2^{t+m} terms, corresponding to $x_1, \dots, x_m, y_1, \dots, y_t \in \{0, 1\}$. Within each of the 2^{t+m} integration regions in (7.17), the function $\zeta_m(z_1, \dots, z_m) = (y_1 + \dots + y_t)/t$ is constant (because the region is contained in a single $R_{m, (y_1, \dots, y_t)}$) and $\partial \lambda_m(z_1, \dots, z_m) / \partial p$ is bounded and continuous (because the region is contained in a single H_{x_1, \dots, x_m}). The function $\lambda_m(z_1, \dots, z_m)$ is also bounded and continuous. Therefore Leibniz's rule can be applied to each integral in (7.17), that

is,

$$\frac{\partial \phi_m(p)}{\partial p} = \sum_{x_1, \dots, x_m, y_1, \dots, y_t} \int_{H_{x_1, \dots, x_m} \cap R_{m, (y_1, \dots, y_t)}} \zeta_m(z_1, \dots, z_m) \frac{\partial \lambda_m(z_1, \dots, z_m)}{\partial p} dz_1 \cdots dz_m. \quad (7.18)$$

The uniform convergence of $\sum_{m=1}^{\infty} \partial \phi_m(p)/\partial p$ is easily obtained from (7.18). The term $r - mp$ in (7.16) can be bounded as $|r - mp| < m$. In addition, for $p \in (\varepsilon, 1 - \varepsilon)$ the term $p(1 - p)$ is lower-bounded by $\varepsilon(1 - \varepsilon)$. Therefore

$$\left| \frac{\partial \lambda_m(z_1, \dots, z_m)}{\partial p} \right| \leq \frac{|r - mp|}{p(1 - p)} \lambda_m(z_1, \dots, z_m) < \frac{m}{\varepsilon(1 - \varepsilon)} \lambda_m(z_1, \dots, z_m). \quad (7.19)$$

In addition, $|\zeta_m(z_1, \dots, z_m)| \leq 1$. Using these two inequalities into (7.18) and taking into account (7.13),

$$\begin{aligned} \left| \frac{\partial \phi_m(p)}{\partial p} \right| &< \frac{m}{\varepsilon(1 - \varepsilon)} \sum_{x_1, \dots, x_m, y_1, \dots, y_t} \int_{H_{x_1, \dots, x_m} \cap R_{m, (y_1, \dots, y_t)}} \lambda_m(z_1, \dots, z_m) dz_1 \cdots dz_m \\ &= \frac{m}{\varepsilon(1 - \varepsilon)} \int_{R_m} \lambda_m(z_1, \dots, z_m) dz_1 \cdots dz_m = \frac{m}{\varepsilon(1 - \varepsilon)} \Pr[M = m]. \end{aligned} \quad (7.20)$$

Let $\Phi_m = m \Pr[M = m]/(\varepsilon(1 - \varepsilon))$. By assumption $E[M]$ is finite, and thus $\sum_{m=1}^{\infty} \Phi_m = E[M]/(\varepsilon(1 - \varepsilon))$ is finite too. Since $|\partial \phi_m(p)/\partial p| < \Phi_m$ with $\sum_{m=1}^{\infty} \Phi_m$ finite, the Weierstrass M -test (Borden, 1998, corollary 10.2.4) establishes that $\sum_{m=1}^{\infty} \partial \phi_m(p)/\partial p$ converges uniformly on $(\varepsilon, 1 - \varepsilon)$. \square

Lemma 7.2. *The estimator \hat{s} given by (7.11), based on the sequential procedure \mathcal{P} , satisfies the following (sequential Cramér-Rao) bound for $p \in (0, 1)$,*

$$\text{Var}[\hat{s}] \geq (f'(p))^2 \frac{p(1 - p)}{t E[N]}, \quad (7.21)$$

where $E[N]$ is the average number of inputs used for generating each Y_j .

Proof. It suffices to prove that (7.21) holds for $p \in (\varepsilon, 1 - \varepsilon)$ with $\varepsilon > 0$ arbitrary. This will be done using Wolfowitz's extension of the Cramér-Rao bound to the sequential case (Wolfowitz, 1947), which particularized to \mathcal{P} and \hat{s} as previously defined will give (7.21).

Consider $\varepsilon > 0$. It will be assumed that $E[N]$ is finite, for otherwise the right-hand side of (7.21) is 0. The desired result will hold if the five regularity conditions enounced by Wolfowitz (1947, section 3) are satisfied. The first condition specifies that p must belong to an open interval. This is indeed the case, as $p \in (\varepsilon, 1 - \varepsilon)$.

The second regularity condition requires that $\partial \lambda(z)/\partial p$ exist for all p and almost all z , and that $E[\partial \log \lambda(Z)/\partial p] = 0$ and $E[(\partial \log \lambda(Z)/\partial p)^2] > 0$ for all $p \in (\varepsilon, 1 - \varepsilon)$, where Z is a generic variable from the sequence Z . This easily follows from (7.12) by computing

$$\frac{\partial \lambda(z)}{\partial p} = \begin{cases} -1 & \text{if } z \in (0, 1) \\ 1 & \text{if } z \in (1, 2). \end{cases} \quad (7.22)$$

and

$$\frac{\partial \log \lambda(z)}{\partial p} = \begin{cases} -1/(1 - p) & \text{if } z \in (0, 1) \\ 1/p & \text{if } z \in (1, 2). \end{cases} \quad (7.23)$$

Since $\Pr[Z \in (0, 1)] = 1 - p$ and $\Pr[Z \in (1, 2)] = p$, from (7.23) it stems that $E[\partial \log \lambda(z)/\partial p] = 0$ and

$$E \left[\left(\frac{\partial \log \lambda(z)}{\partial p} \right)^2 \right] = \frac{1}{1-p} + \frac{1}{p} = \frac{1}{p(1-p)}, \quad (7.24)$$

which is strictly positive as required.

The third condition requires that, for $m \in \mathbb{N}$ and for variables Z_1, \dots, Z_m belonging to sequence Z ,

$$E \left[\left(\sum_{j=1}^m \left| \frac{\partial \log \lambda(Z_j)}{\partial p} \right| \right)^2 \right] \quad (7.25)$$

exists for all p in the considered interval. This is satisfied because, according to (7.23), $|\partial \log \lambda(Z_j)/\partial p|$ is upper-bounded by $1/\varepsilon$ for $p \in (\varepsilon, 1 - \varepsilon)$.

The fourth condition states that $|\zeta_m(z_1, \dots, z_m) \partial \lambda_m(z_1, \dots, z_m)/\partial p|$ be bounded by a measurable function of z_1, \dots, z_m with finite integral on R_m . This clearly holds because $|\zeta_m(z_1, \dots, z_m)| \leq 1$ and, from (7.16), $|\partial \lambda_m(z_1, \dots, z_m)/\partial p| < m$.

Lastly, the fifth regularity condition postulates the uniform convergence of the series $\sum_{m=1}^{\infty} \partial \phi_m(p)/\partial p$. As the sequential estimator \hat{s} uses t repetitions of the Bernoulli factory \mathcal{B}' , the average number of inputs $E[M]$ is given by

$$E[M] = t E[N] \quad (7.26)$$

and is thus finite. Therefore Lemma 7.1 establishes the required uniform convergence.

Since the regularity conditions are satisfied, the main inequality for the single-parameter case in Wolfowitz (1947, equation (4.5)) holds, namely

$$\text{Var}[\hat{s}] = \frac{(\text{d}E[\hat{s}]/\text{d}p)^2}{E[M] E \left[(\partial \log \lambda(z)/\partial p)^2 \right]}. \quad (7.27)$$

The estimator (7.11) is unbiased. Thus

$$\frac{\text{d}E[\hat{s}]}{\text{d}p} = \frac{\text{d}s}{\text{d}p} = f'(p). \quad (7.28)$$

Substituting (7.24), (7.26) and (7.28) into (7.27) yields (7.21). \square

The proof of the theorem uses the following upper bound on the variance of \hat{s} , which is obtained from (7.11),

$$\text{Var}[\hat{s}] = \frac{s(1-s)}{t} < \frac{s}{t}, \quad (7.29)$$

as well as the lower bound on $\text{Var}[\hat{s}]$ given by Lemma 7.2.

The function f satisfies (3.1), and since it is differentiable, (3.2) also holds. Thus $f(p)/p^a$ and $f'(p)p^{1-a}$ can be approximated by b and ab respectively for p small. In addition, since $1/s - t = 1/s - \lfloor 1/s \rfloor \in [0, 1)$ and $\lim_{p \rightarrow 0} s = \lim_{p \rightarrow 0} f(p) = 0$, it follows that $st \rightarrow 1$ for $p \rightarrow 0$, that is, st can be approximated by 1 for p small.

Specifically, given ε there exists $\delta < 1/2$ such that the following three inequalities hold for all $p \in (0, \delta)$

$$\left| \frac{f(p)}{p^a} - b \right| < \varepsilon, \quad (7.30)$$

$$|f'(p)p^{1-a} - ab| < \varepsilon, \quad (7.31)$$

$$|st - 1| < \varepsilon \quad (7.32)$$

or equivalently

$$(b - \varepsilon)p^a < f(p) < (b + \varepsilon)p^a, \quad (7.33)$$

$$(ab - \varepsilon)p^{a-1} < f'(p) < (ab + \varepsilon)p^{a-1}, \quad (7.34)$$

$$\frac{1 - \varepsilon}{s} < t < \frac{1 + \varepsilon}{s}. \quad (7.35)$$

Since $s = f(p)$, combining (7.29), (7.33) and (7.35) gives

$$\text{Var}[\hat{s}] < \frac{s^2}{1 - \varepsilon} < \frac{(b + \varepsilon)^2 p^{2a}}{1 - \varepsilon}. \quad (7.36)$$

On the other hand, using (7.34) into inequality (7.21) from Lemma 7.2 gives, for $p \in (0, \delta)$,

$$\text{Var}[\hat{s}] > \frac{(ab - \varepsilon)^2 p^{2a-1} (1 - p)}{t \mathbb{E}[N]}. \quad (7.37)$$

Substituting $t = 1/s = 1/f(p)$ into (7.37) and using (7.33) yields

$$\text{Var}[\hat{s}] > \frac{(ab - \varepsilon)^2 (b - \varepsilon) p^{3a-1} (1 - p)}{\mathbb{E}[N]}. \quad (7.38)$$

Inequalities (7.36) and (7.38) imply that, for $p \in (0, \delta)$,

$$\frac{(b + \varepsilon)^2 p^{2a}}{1 - \varepsilon} > \frac{(ab - \varepsilon)^2 (b - \varepsilon) p^{3a-1} (1 - p)}{\mathbb{E}[N]}. \quad (7.39)$$

Since $p < \delta < 1/2$, the factor $1 - p$ is lower-bounded by $1/2$. Using this and rearranging terms, (7.39) becomes

$$\mathbb{E}[N] > \frac{(ab - \varepsilon)^2 (b - \varepsilon) (1 - \varepsilon)}{2(b + \varepsilon)^2} p^{a-1}. \quad (7.40)$$

Making use of (7.33) again,

$$\mathbb{E}[N] > \frac{(ab - \varepsilon)^2 (b - \varepsilon) (1 - \varepsilon)}{2(b + \varepsilon)^3} \frac{f(p)}{p}. \quad (7.41)$$

Taking $\varepsilon = \min\{ab/2, 1/2\}$, which ensures that $\varepsilon \leq ab/2 < b/2$ and $\varepsilon \leq 1/2$, the first fraction in (7.41) is easily seen to be lower-bounded by $a^2/108$. Thus

$$\mathbb{E}[N] > \frac{a^2}{108} \frac{f(p)}{p} \quad \text{for all } p \in (0, \delta), \quad (7.42)$$

which implies that $\mathbb{E}[N]$ is $\Omega(f(p)/p)$. \square

7.5. Proof of Theorem 4.1

The following result about geometric random variables will be needed.

Lemma 7.3. *Geometric random variables have exponential tails.*

Proof. Given $\theta \in (0, 1)$, consider a random variable R with $\Pr[R = r] = \theta(1 - \theta)^{r-1}$, $r \in \mathbb{N}$. Computing $\Pr[R \geq r + 1] = (1 - \theta)^r$ shows that (1.1) is satisfied with $A = 1$, $\beta = 1 - \theta < 1$. \square

Steps 2–4 of Algorithm 4.1 (which are the same as in Algorithm 2.1 except step 3) form an (outermost) loop on i which is repeated until the exit condition in step 4 is met. For each iteration of this loop, Algorithm 2.1 uses one input from X ; whereas Algorithm 4.1 uses that input plus additional ones that are needed for generating V_i , as specified by steps 3.1–3.3. For a given i , the number of iterations of the loop on j formed by steps 3.2 and 3.3 is a geometric random variable K_i with parameter $1/2$, and thus $E[K_i] = 2$. For each j , the number of blocks of 2 inputs required within step 3.2 is a geometric random variable $L_{i,j}$ with parameter $2p(1 - p)$, and thus $E[L_{i,j}] = 1/(2p(1 - p))$. Let

$$L_i = L_{i,1} + \cdots + L_{i,K_i}. \quad (7.43)$$

The variables $L_{i,j}$ are i.i.d. and independent from K_i , and therefore (Papoulis, 1991, page 194) $E[L_i] = E[K_i]E[L_{i,1}] = 1/(p(1 - p))$. Thus, for each i Algorithm 4.1 uses one input from X in step 2 (like Algorithm 2.1 does), plus $1/(p(1 - p))$ blocks of 2 inputs on average in steps 3.1–3.3. Consequently, Algorithm 4.1 uses on average $1 + 2/(p(1 - p))$ as many inputs as Algorithm 2.1 does. This establishes (4.1).

By Lemma 7.3, the variables $L_{i,j}$ as well as K_i have exponential tails; and then proposition 12 of Nacu & Peres (2005) guarantees that L_i has an exponential tail. For each i , the iteration formed by steps 2–4 of Algorithm 4.1 requires $1 + 2L_i$ inputs. The total number of iterations of this outermost loop coincides with the number of inputs of Algorithm 2.1, which has an exponential tail as established by Theorem 4.1. Since $1 + 2L_i$ also has an exponential tail, applying proposition 12 of Nacu & Peres (2005) again shows that the total number of inputs used by Algorithm 4.1 has an exponential tail. \square

7.6. Proof of Proposition 5.1

Consider functions $f_1(p) = 1 - \sum_{i=1}^{\infty} c_{1,i}(1 - p)^i$ and $f_2(p) = 1 - \sum_{j=1}^{\infty} c_{2,j}(1 - p)^j$ that satisfy (2.1)–(2.3) and (3.1); and let f , g and h be defined as in (5.1)–(5.3).

Regarding f , identifying coefficients in

$$f(p) = f_2(f_1(p)) = 1 - \sum_{j=1}^{\infty} c_{2,j} \left(\sum_{i=1}^{\infty} c_{1,i}(1 - p)^i \right)^j = 1 - \sum_{k=1}^{\infty} c_k(1 - p)^k \quad (7.44)$$

it is seen that $c_k \geq 0$ for $c_{1,i}, c_{2,j} \geq 0$. Also, $\lim_{p \rightarrow 0} f(p) = 0$ because $\lim_{p \rightarrow 0} f_1(p) = \lim_{p \rightarrow 0} f_2(p) = 0$; and thus $\sum_{k=1}^{\infty} c_k = 1$. As for condition (3.1), by assumption

$$\lim_{p \rightarrow 0} f_1(p)/p^{a_1} = b_1, \quad \lim_{p \rightarrow 0} f_2(p)/p^{a_2} = b_2 \quad (7.45)$$

with $a_1, a_2 \in (0, 1)$ and $b_1, b_2 \in (0, \infty)$. This implies that

$$\lim_{p \rightarrow 0} \frac{f_2(f_1(p))}{(b_1 p^{a_1})^{a_2}} = \lim_{p \rightarrow 0} \frac{f_2\left(\frac{f_1(p)}{p^{a_1}} p^{a_1}\right)}{(b_1 p^{a_1})^{a_2}} = \lim_{p \rightarrow 0} \frac{f_2(b_1 p^{a_1})}{(b_1 p^{a_1})^{a_2}} = b_2. \quad (7.46)$$

Therefore f satisfies (3.1) with $a = a_1 a_2 \in (0, 1)$, $b = b_1^{a_2} b_2 \in (0, \infty)$.

As for g , writing

$$f(p) = 1 - (1 - f_1(p))(1 - f_2(p)) = 1 - \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} c_{1,i} c_{2,j} (1-p)^{i+j} = 1 - \sum_{k=1}^{\infty} c_k (1-p)^k \quad (7.47)$$

and identifying coefficients again makes it clear that $c_k \geq 0$. On the other hand, it is easily seen that $\lim_{p \rightarrow 0} f(p) = 0$, which implies that $\sum_{k=1}^{\infty} c_k = 1$. Finally, since

$$f(p) = 1 - (1 - f_1(p))(1 - f_2(p)) = f_1(p) + f_2(p) - f_1(p)f_2(p), \quad (7.48)$$

condition (3.1) is seen to hold with $a = \min_{i=1,2} \{a_i\}$ and $b = b_{\arg \min_{i=1,2} \{a_i\}}$ for $a_1 \neq a_2$, $b = b_1 + b_2$ for $a_1 = a_2$, where a_1, b_1, a_2, b_2 are as in (7.45).

The proof for h is similar. \square

References

- BORDEN, R. S. (1998). *A Course in Advanced Calculus*. Dover.
- FLEMING, W. (1977). *Functions of Several Variables*. Springer-Verlag, 2nd ed.
- HUBER, M. (2016). Nearly optimal Bernoulli factories for linear functions. *Combinatorics, Probability and Computing* **25**, 577–591.
- KEANE, M. & O'BRIEN, G. L. (1994). A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation* **4**, 213–219.
- ŁATUSZYŃSKI, K., KOSMIDIS, I., PAPASILIOPOULOS, O. & ROBERTS, G. O. (2011). Simulating events of unknown probabilities via reverse time martingales. *Random Structures and Algorithms* **38**, 441–452.
- NACU, Ş. & PERES, Y. (2005). Fast simulation of new coins from old. *Annals of Applied Probability* **15**, 93–115.
- PAPOULIS, A. (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd ed.
- VON NEUMANN, J. (1951). Various techniques used in connection with random digits. *National Bureau of Standards Applied Mathematics* **12**, 36–38.
- WÄSTLUND, J. (1999). Functions arising by coin flipping. <http://www.math.chalmers.se/~wastlund/coinFlip.pdf>. Chalmers University of Technology.
- WOLFOWITZ, J. (1947). The efficiency of sequential estimates and Wald's equation for sequential processes. *Annals of Mathematical Statistics* **18**, 215–230.